AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1.   (Original) An executable code check system comprising:

an input component that receives an object file having an embedded specification; and,

a checker that employs the specification to facilitate static checking of the object file, the checker providing information if a fault condition is determined.

2.   (Original) The system of claim 1, the checker further removing the embedded specification from the object file.

3.   (Original) The system of claim 1, the specification comprising information associated with a method that performs at least one of allocation and release of a resource.

4.   (Original) The system of claim 1, the specification comprising information associated with an order in which methods of an object can be called.

5.   (Original) The system of claim 4, wherein method order is constrained by specifying a finite state machine in which the states have symbolic names and transitions between states are labeled with method names.

6.   (Original) The system of claim 1, the specification comprising a state-machine protocol wherein a method specifies a pre-state and a post-state.

7.   (Original) The system of claim 1, the specification comprising information associated with a transition of a finite state machine.

8.   (Original) The system of claim 1, the specification comprising at least one of a rule using an interface, system resource management, order of method calls and formatting of a string parameter.

9.   (Original) The system of claim 1, the object file being based, at least in part, upon a language that compile to Common Language Runtime.

10.   (Currently Amended) The system of claim 1, the object file being based, at least in part, upon at least one of C#, ~~Visual Basic.net~~ VISUAL BASIC.NET and Managed C++.

11.   (Original) The system of claim 1, the specification comprising information associated with a state-machine protocol.

12.   (Original) The system of claim 1, the specification comprising an attribute associated with at least one of a field and a parameter providing information associated with whether or not the at least one of a field and a parameter can be aliased.

13.   (Original) The system of claim 1, wherein the specification facilitates modeling of a heap modeling.

14.   (Original) The system of claim 13, the checker employing an algorithm that performs a data flow analysis over the heap model comprising a typing environment and a set of capabilities.

15.   (Original) An executable code check system comprising:
        an input component that receives an object file;

a checker that employs a specification associated with the object file to facilitate static checking of the object file, the checker providing information if a fault condition is determined, the specification stored in a specification repository.

(Original) The system of claim 15, further comprising the specification repository.

17.   (Original) A method of facilitating static checking of executable code comprising:
       receiving executable code with an embedded specification;
       statically applying the specification to the executable code;
       determining whether a fault condition exists based, at least in part, upon the
statically applied specification; and,
       providing information associated with the fault condition, if a fault condition is
determined to exist.

18.   (Original) The method of claim 17, further comprising removing the embedded specification from the executable code.

19.   (Original) A computer readable medium having stored thereon computer executable instructions for carrying out the method of claim 17.

20.   (Original) A method of facilitating static checking of executable code comprising:
       receiving executable code;
retrieving a specification associated with the executable code;
       statically applying the specification to the executable code;
       determining whether a fault condition exists based, at least in part, upon the
statically applied specification; and,
       providing information associated with the fault condition, if a fault condition is
determined to exist.

21.   (Original) A computer readable medium having stored thereon computer executable instructions for carrying out the method of claim 20.

22.   (Original) A data packet transmitted between two or more computer components that facilitates static checking of executable code, the data packet comprising:

executable code having an embedded specification, the embedded specification providing information to be employed to statically check the executable code.

23.   (Original) A data packet transmitted between two or more computer components that facilitates static checking of executable code, the data packet comprising:

a specification that provides information to be employed to statically check the executable code.

24.   (Original) A computer readable medium storing computer executable components of an executable code check system comprising:

an input component that receives an object file having an embedded specification; and,

a checker component that employs the specification to facilitate static checking of the object file, the checker providing information if a fault condition is determined.

25.   (Original) An executable code check system comprising:

means for receiving an object file having an embedded specification; and,

means for statically checking the object file based, at least in part, upon the embedded specification and determining if a fault condition exists; and,

means for providing information if a fault condition is determined to exist.